

Postagem do Trabalho no Moodle – 16/04/2017 (domingo)
Defesas do Trabalho: 17 e 24/04/2017

1º TRABALHO PRÁTICO – ETAPA 1

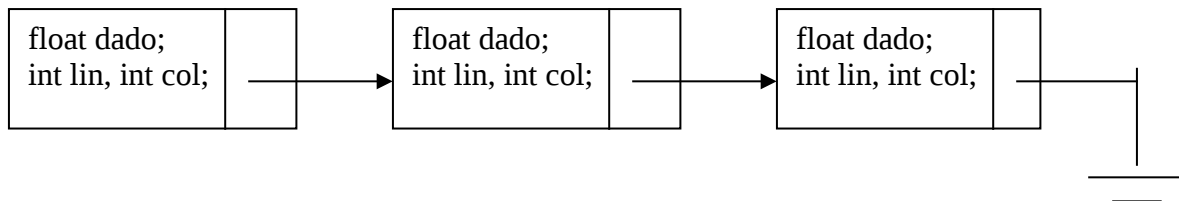
Criar um programa em C que manipule matrizes esparsas, utilizando lista linear encadeada.

OBS: Uma matriz é considerada esparsa quando a maioria dos seus dados são zeros.

Exemplo de Matriz Esparsa

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3.5
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	5	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	0

Em caso da matriz ser muito grande não é vantajoso, em termos de memória, armazenar todos os dados, até porque a maioria são zeros. Então o objetivo do trabalho é criar uma lista encadeada e armazenar somente os dados diferentes de zero, bem como a linha e a coluna onde esses dados estão localizados.



Mas, no entanto, o programa deverá imprimir na saída a matriz inteira, tanto os dados diferentes de zero como os zeros.

O programa deverá ser formado pela seguinte struct:

```
typedef struct nodo
{
    float dado;
    int lin, col;
    struct nodo *prox;
}Matriz_Esparsa;
```

E pelas seguintes funções:

- Uma função que faz a alocação de memória para cada nodo criado para uma lista encadeada;
- Uma função que insere na lista encadeada um nodo alocado;
- Uma função que busca os dados em uma lista encadeada;

- Uma função que libera da memória uma lista encadeada;
- Uma função que lê os dados da matriz, via teclado, e insere na lista encadeada somente os dados diferentes de zero;
- Uma função que soma duas matrizes;
- Uma função que subtrai duas matrizes;
- Uma função que multiplica duas matrizes;
- Uma função que gera a matriz transposta;
- Uma função que imprime todos os dados da matriz, inclusive os zeros;
- Uma função que imprime os elementos da diagonal principal, inclusive os zeros caso existam.

Observações:

- O usuário deve informar a dimensão das matrizes.
- Considerar as regras de operações com matrizes para implementar a soma, subtração, multiplicação, transposta e diagonal principal de matrizes.
- As listas encadeadas devem ser passadas como parâmetros das funções.
- O usuário poderá fazer quantas operações desejar com as matrizes.

Avaliação:

- **(2.0 ponto)** Implementação de todas as funções
- **(1.5 ponto)** programa rodando
- **(1.5 pontos)** defesa individual

ATENÇÃO:

1. Equipes de no máximo dois alunos; em caso de cópias de trabalhos todas as equipes envolvidas receberão nota zero.
2. Questões sobre o trabalho podem ser solicitadas na 1ª prova teórica.